Database Tuning, ITU, Spring 2007

Rasmus Pagh

February 5, 2007

Project – deliverable 1

This is the first part of the Database Tuning project. The project is to be carried out in groups of 3-4 students; the members of each group are announced on the course home page. It is up to the members of your group to agree on how to work together (meeting times, etc.) The project will be structured in 5 parts, where the first 4 have a deliverable to be handed in during the course. The final project report will contain revised versions of these deliverables, with the possibility to work on and write about additional (performance) aspects of the development project.

Deadline for deliverable 1: February 15, 23.59 Danish time.

Purpose

The Database Tuning project is a database development project with focus on performance. The performance will be tested by programs that simulate use of the database, provided by the teachers of the course. The first deliverable builds upon skills acquired in an introductory database course, and aims at getting an understanding of the problem domain addressed by the database by constructing an initial E/R diagram.

Time plan for deliverable 1

February 6. Project introduction. Case description, question/answer session.

February 13. Office hours for project supervisor (Milan Ruzic).

February 15. Hand-in by e-mail (pdf file) to `milan@itu.dk`

February 20. Feedback (individually for each group).

Time plan future deliverables

Deliverable 2: Start February 20, **hand-in March 8**, feedback March 15.

Deliverable 3: Start March 15, **hand-in March 29**, feedback by e-mail around April 10.

Deliverable 4: Start April 10, **hand-in april 26**, feedback May 1.

Extras + final report: Start May 1, **hand in May 23**, feedback at exam!

Case description

The following is only a partial description of the case. To obtain a full understanding it is probably necessary to ask questions to the recipients of the result of the project (your teachers).

Viking Xpress Logistics is a newly started Danish delivery service. Its customers order door-to-door delivery of parcels nationwide. The business idea of VXL is to provide a very flexible service that will allow customers to hand over their post at any time, and still ensure that parcels are routed in a fast and cost-effective way to its destination. The routing of parcels will be handled by software by the little-known German company *Pfadfinder AG*. The aim of this project is to create a database that will allow VXL to keep track of their parcels and other business/organizatorial information.

Traditional logistics companies gather all post in large hubs on a daily basis, and ship everything from there. The idea behind VXL is to provide a more flexible (and often faster) routing of parcels by using an infrastructure similar to public transportation. For example, a parcel traveling from the IT University of Copenhagen to Kronborg Castle in Elsinore might first be shipped by car to Taarnby Station, then later picked up and transported by train to Elsinore train station, and after a short waiting time finally be shipped to Hamlet's castle by a bike courier. To make this work, VXL has safe storage containers set up at railroad stations and thousands of other places across the country, where the parcels can be placed while waiting to be carried further. Some VXL vehicles travel according to a fixed schedule, and some can be reserved to carry a particular parcel (mainly at route end points). VXL's customers have client software installed that makes queries to the database on routing possibilities, and reserves space on vehicles for a particular parcel. Pricing may vary according to what route is chosen.

The VXL database must contain:

- All road segments in Denmark, including name of the street, postal code, and street numbers.
- All endpoints of road segments, including their geographical coordinates. (Several road segments may share an endpoint.)
- Name and geographical location (road segment and street number) of each customer.
- Routing information for every parcel (travel plan), including IDs of vehicles, pickup and delivery address, customer ID, and shipping price.
- A complete history of all things that happen to parcels, such that customers can check the locations of their parcels at any time. This includes depositing in and pickup from safe storage containers.
- A GPS based system informs the VXL headquarters about the position of each of its vehicles every minute. The geographical position is translated into a road segment that is stored in the database along with the corresponding time stamp and vehicle number. It is also recorded by how much the vehicle is behind schedule (if applicable).
- For every vehicle, the type and model name is stored, along with information on date of purchase, maximum speed, carrying capacity (kilograms and liters), and mileage. Milage is reported by drivers on a daily basis.
- The schedule of every vehicle. For some vehicles the schedule is fixed weekly by the VXL headquarters, while others are bookable by the Pfadfinder software. (This information is stored in the database.) An entry in the schedule consists of a starting point and time, and an ending time and point, with no stop in between.
- Name and ID of every driver. It is recorded in what time periods a driver is working, and in what vehicles. (Each vehicle has a single driver.)

- The drivers are organized by county, and the counties are organized by region. It must be possible to identify the drivers that are based in a particular county or region. (There is no requirement to keep track of the county that a vehicle is in.)

The database must ensure referential integrity for foreign key references. However, other constraints such as the fact that a driver cannot be driving two vehicles at the same time are supposed to be enforced by the software that adds data to the database.

To be handed in

A pdf file with the following:

- An E/R diagram for the VXL database, plus written explanation where necessary. You must state what E/R notation is used (preferably Chen or the one in RG).
- DDL statements for creating the corresponding relations in Oracle, including integrity constraints. For now you may make any reasonable assumption on the data types (e.g. maximal length of strings). More information will be provided later.